The Fast Decoding of Reed-Solomon Codes Using Number Theoretic Transforms

I. S. Reed and L. R. Welch
University of Southern California

T. K. Truong
DSN Systems Engineering Office

It is shown that Reed-Solomon (RS) codes can be encoded and decoded by using a fast Fourier transform (FFT) algorithm over finite fields. A Fourier-like transform is defined over finite fields of type $I_{F_n}(\sqrt[8]{2})$, where F_n is a Fermat prime for $n \leq 4$. The field $I_{F_n}(\sqrt[8]{2})$ is used to extend the length of the original Fermat number transforms by a factor of 8. The arithmetic utilized to perform these transforms over the field of type $I_{F_n}(\sqrt[8]{2})$ requires only integer additions, circular shifts and a minimum number of integer multiplications by powers of $\sqrt[8]{2}$. The computing time of this transform encoder-decoder for RS codes is less than the time of the standard method for RS codes.

More generally, the field GF(q) is also considered, where q is a prime of the form $K \times 2^n + 1$ and K and n are integers. GF(q) can be used to decode very long RS codes by an efficient FFT algorithm with an improvement in the number of symbols. The arithmetic needed for these more general transforms requires only slightly modified binary integer additions and multiplications.

Transforms can be defined also over the Galois field $GF(q^2)$, a finite field analogous to the complex number field, where $q = 2^p - 1$ is a Mersenne prime. The arithmetic needed for this case requires integer complex multiplications mod q and additions mod q.

It is shown in this paper that a radix-8 FFT algorithm over $GF(q^2)$ can be utilized to encode and decode very long RS codes with a large number of symbols. For eight symbols in $GF(q^2)$, this transform over $GF(q^2)$ can be made simpler than any other known number theoretic transform with a similar capability. Of special interest is the decoding of a 16-tuple RS code with four errors.

I. Introduction

Recently Gore (Ref. 1) extended Mandelbaum's methods (Ref. 2) for decoding Reed-Solomon (RS) codes (Ref. 3). He proposed the usage of a finite field transform over $GF(q^n)$, where q is a prime and n is an integer, for decoding RS codes. Michelson (Ref. 4) has implemented Mandelbaum's algorithm and showed that the decoder, using the transform over $GF(q^n)$, is faster than a more standard decoder (Ref. 5). The first disadvantage of the transform method over $GF(q^n)$ is that the transform length is an odd number, so that the most efficient FFT algorithm cannot be used to yield a fast transform decoder. The second disadvantage is that the arithmetic required to perform these transforms over $GF(q^n)$ still requires a substantial number of multiplications in $GF(q^n)$. The arithmetic used to implement this transform was performed in the extended field, $GF(q^n)$.

Schonhage and Strassen (Ref. 6) defined Fourier-like transforms over the ring of integers modulo the Fermat number $2^{2^n} + 1$ to yield convolutions for performing fast integer multiplications. Rader (Ref. 7) proposed transforms over rings of integers modulo both Mersenne and Fermat numbers that can be used to compute error-free convolutions of real integer sequences.

Agarwal and Burrus (Refs. 8 and 9) extended Rader's Fermat number theoretic transform by using the generator $\alpha = \sqrt{2}$ for the transform, rather than $\alpha = 2$. In this case the usual FFT algorithm can be used to calculate transforms with as many as 2^{n+2} points of integer data. This transform was shown to be over the residue classes of quadratic integers $I_{F_n}(\sqrt[2]{2})$, where $\sqrt{2}$ is a root of $x^2 - 2 = 0$ and I_{F_n} denotes the set of integers mod F_n (Ref. 10).

McClellan (Ref. 11) has realized recently the hardware for the Fermat number theoretic transforms. He showed that the arithmetic used to perform these transforms requires only integer additions and circular shifts. The primary advantage of the Rader transform is that multiplications by powers of two are performed by simple bit rotations. Of course, this advantage must be weighed against the difficulty of the numeric constraints relating word length, length of sequence d, and the compositeness of d, imposed by the choices of Mersenne prime and Fermat numbers.

Recently, the authors (Refs. 12 and 13) extended the number theoretic transform (NTT) to a complex integer field by taking transforms over a Galois field $GF(q^2)$, where q is a Mersenne prime. This field is analogous to the field of complex numbers. Such a complex number theoretic transform (CNT) offers more choices in transform length than can be obtained by other methods for the computing fast transform of the complex numbers. The arithmetic used to perform this transform requires integer complex multiplications and additions, mod q. In Ref. 14 it was shown that the binary arithmetic in $GF(q^2)$ is simpler than complex number arithmetic. For example the components of the eighth roots of unity in $GF(q^2)$ are fixed powers of 2. This latter fact was used to develop a fast radix–8 FFT algorithm over $GF(q^2)$. The transforms over $GF(q^2)$ were extended also to operate over the direct sum of Galois fields (Ref. 15). Such transforms can be used to compute transforms with improved dynamic range.

It was proposed (Ref. 16) also that number theoretic transforms could be defined in the Galois field $GF(q)$, where the prime q was of form $q = k \times 2^n + 1$, where k and n are integers. For this class of primes, the FFT algorithm can be utilized to realize transforms of integers that are not quite as fast as the Fermat number transforms. However, such transforms offer a substantial variety of transform and word lengths beyond what is possible with the Fermat transforms of Schonhage, Strassen, and Rader.

The arithmetic used to perfom the FFT over $GF(q)$ requires only slightly modified binary integer additions and multiplications. It should be noted that in Ref. 16, a method to perform arithmetic modulo $k \times 2^n + 1$ is developed specifically for the case $k = 3$. It was shown (Ref. 17) that a radix–2 FFT over $GF(q)$ is slightly faster than the efficient algorithm (Ref. 18) for the conventional FFT of real data when programmed on a PDP-10 computer. This speed could be considerably improved on computer hardware appropriately specialized to perform modulo q arithmetic.

Recently, Justesen (Ref. 19) proposed that transforms over fields of Fermat primes can be used to encode and decode RS codes. Since $\sqrt{2} \in GF(F_n)$ for $n = 2, 3, 4$ (see Ref. 10) is an element of order 2^{n+2} in $GF(F_n)$, the RS code of as many as 2^{n+2} symbols can be generated in $GF(F_n)$. Hence, using an argument similar to Gore's transform decoding method, mentioned above, the Fermat number theoretic transform is used to decode RS codes. Since the arithmetic in this new transform decoder is performed in $GF(F_n)$, such a number theoretic trans-

form decoder for RS codes can handle as many as 2^{n+2} symbols for $n = 2, 3, 4$. Encoding, and decoding can be accomplished faster and simpler than any other known standard decoder for RS codes of the same symbol range.

To treat longer RS codes in $GF(F_n)$, the transform is extended here to the finite field of type $I_{F_n}(\sqrt[8]{2})$, where $\sqrt[8]{2}$ is a root of the polynomial $P(x) = x^8 - 2$ over $GF(F_n)$ and I_{F_n} denotes the set of integers modulo F_n. If F_n is a Fermat prime, then $I_{F_n} = GF(F_n)$. The field $I_{F_n}(\sqrt[8]{2})$ is obtained by taking the residue classes of polynomials modulo $P(x)$. That is,

$$I_{F_n}(\sqrt[8]{2}) = \{a + b(\sqrt[8]{2}) + c(\sqrt[8]{2})^2 + d(\sqrt[8]{2})^3 + e(\sqrt[8]{2})^4 + f(\sqrt[8]{2})^5 + g(\sqrt[8]{2})^6 + h(\sqrt[8]{2})^7 \mid a, b, c, d, e, f, g, h \epsilon GF(F_n)\}$$

It will be shown that 2 is an octadic residue of a Fermat prime F_n for $n = 3, 4$. Thus, $I_{F_n}(\sqrt[8]{2})$ is a field of F_n elements isomorphic to $GF(F_n)$. The transform over $I_{F_n}(\sqrt[8]{2})$ extends the length of Rader's original Fermat number theoretic transform by a factor of 8. The arithmetic used to perform this transform requires only integer additions, circular shifts and a minimum number of integer multiplications by powers of $\sqrt[8]{2}$.

To decode very long RS codes over $GF(F_n)$ (from Refs. 9 and 10), one can use the fact that 3 is a primitive element in $GF(2^{2^n} + 1)$. Thus a FFT over $GF(F_n)$ can be used to decode a 2^{2^n}-tuple RS code. The arithmetic used to perform this transform requires integer multiplications by powers of 3 and integer additions mod F_n.

Since the Fermat primes F_n exist only for $n \leq 4$, the dynamic range of the transforms associated with these primes is severely limited. To remedy this it may be possible to use transforms over the direct sum of Galois fields, $GF(F_n)$ to decode RS codes with an improved number of symbols.

A special case of the radix-8 FFT over $GF(q^2)$ where $q = 2^p - 1$ is a Mersenne prime is developed in some detail to encode and decode a very long nonsystematic RS code with a large number of symbols. Recall that the 8th root of unity in $GF(q^2)$ is $\pm 2^{(p-1)/2}(1 + \hat{i})$, where p is a prime. Hence, the arithmetic used to perform 8-point transforms requires only circular shifts and additions. This transform is used to decode a 16-tuple error correcting RS code faster and simpler than any other similar code.

II. A Transform Over $I_{F_n}(\sqrt[8]{2})$ Where F_n is a Fermat Prime

Let q be a prime and let $GF(q^n)$ be the Galois field and suppose that integer d divides $q^n - 1$. Also let the element $\gamma \epsilon GF(q^n)$ generate the cyclic subgroup of d elements, G_d in the multiplicative group of $GF(q^n)$. Then, by Ref. 12, a transform over this subgroup G_d can be defined by

$$A_K = \sum_{n=0}^{d-1} a_n \gamma^{Kn} \qquad \text{for } 0 \leq K \leq d-1 \qquad \text{(1a)}$$

where d divides $q^n - 1$ and $a_n \epsilon GF(q^n)$ for $n = 0, 1, 2, \cdots, d-1$ and the inverse of transform of A_K is

$$a_n = (d)^{-1} \sum_{K=0}^{d-1} A_K \gamma^{-Kn} \qquad \text{(1b)}$$

where (d) denotes the residue of d mod q, and $(d)^{-1}$ is the inverse of (d). In the present case, attention is restricted to $n = 1$. Thus, the transform over $G_d c GF(q)$ can be defined by

$$A_K = \sum_{n=0}^{d-1} a_n \gamma^{Kn} \qquad \text{for } 0 \leq K \leq d-1 \qquad \text{(2a)}$$

where $d \mid q - 1$ and $a_n, A_K \epsilon GF(q)$ and the inverse transform of A_K still holds. That is,

$$a_n = (d)^{-1} \sum_{K=0}^{d-1} A_K \gamma^{-Kn} \qquad \text{for } 0 \leq n \leq d-1 \qquad \text{(2b)}$$

where (d) denotes the residue of d mod q and $(d)^{-1}$ is the inverse of (d).

It is shown in the Appendix that 2 is an octadic residue of a Fermat prime F_n. As a consequence, $\sqrt[8]{2}$ is an element of $GF(F_n)$. Thus, by the same procedure used in the proof of theorem 6 of Ref. 21, $I_{F_n}(\sqrt[8]{2})$ for $n = 3, 4$ is isomorphic to $GF(F_n)$.

If $q = F_n$ is a Fermat Prime, the above transform (Eq. (2)) can be defined in $GF(F_n) \cong I_{F_n}(\sqrt[8]{2})$ for $n = 3, 4$. Since $I_{F_n}(\sqrt[8]{2})$ is isomorphic to $GF(F_n)$ and $(\sqrt[8]{2})^{2^{n+3}} \equiv -1$ mod F_n, then by the theorem 1 of Ref. 12, $\gamma = \sqrt[8]{2}$, is an element of order 2^{n+4} in $I_{F_n}(\sqrt[8]{2})$. Thus the FFT over $I_{F_n}(\sqrt[8]{2})$ can be defined to compute the transform of a sequence of as many as $d = n + 4$ points of integer data. It should be noted that this fact extends the length of Rader's original Fermat number theoretic transform by a factor of 8.

Since $\gamma = \sqrt[8]{2}\ GF(F_n)$ is an element of order 2^{n+4}, it is well known (see for example Ref. 12) that the FFT algorithm over $GF(F_n)$ is composed of $d = n + 4$ stages of computation. The first $d - 3$ stages require only multiplications by the powers of 2, i.e., circular shift. By Ref. 9, $\sqrt{2} = 2^{3 \cdot 2^{n-2}} - 2^{2^{n-2}}$. Thus, the $(d-2)$-st stage requires integer multiplications by powers of $\sqrt[8]{2}$, i.e., circular shifts. Only the last two stages require integer multiplications by powers of $\sqrt[8]{2}$. Hence the number of arithmetic operations used to perform this transform are $d \cdot \log d$ integer additions $((d-3) \cdot \log d + 2 \log d) = (d+1) \log d$ circular shifts and $2 \cdot \log d$ integer multiplications by powers of $\sqrt[8]{2}$. This implies that the FFT over $I_{F_n}(\sqrt[8]{2})$ for $n = 3, 4$ is faster and simpler than any other number theoretic transforms of the same transform length and dynamic range.

III. Fast Decoding of Systematic Reed-Solomon Codes Using the Transform Over $I_{F_n}(\sqrt[8]{2})$

It was shown in the previous section that the field of type $I_{F_n}(\sqrt[8]{2})$ is isomorphic to $GF(F_n)$ for $n = 3, 4$ and the $\alpha = \sqrt[8]{2}\ \epsilon GF(F_n)$ is an element of order 2^{n+4}. A systematic Reed-Solomon code can be specified in $GF(F_n)$ as follows.

Assume the code length for the RS code is $N = 2^{n+4}$ Let a codeword be represented by $f(x)$, a polynomial of degree $N - 1$ over $GF(F_n)$. The generator polynomial of $f(x)$ is defined as

$$g(x) = \sum_{i=1}^{d-1} (x - \alpha^i)$$

where $d = 2^K < N = 2^{n+4}$, $\alpha = \sqrt[8]{2}$, $\alpha^2 = (\sqrt[8]{2})^2, \cdots, \alpha^d = (\sqrt[8]{2})^d$ are the roots of $g(x)$ in $GF(F_n)$. The resultant RS code with N symbols, which is a multiple of the generator polynomial, is composed of $d - 1$ parity check symbols and $n - (d - 1)$ information symbols. d is the minimum distance of the RS code. If t is the number of errors the code will correct, then for an RS code $d = 2t + 1$.

Suppose that the code $f(x) = f_0 + f_1 x + \cdots + f_{N-1} x^{N-1}$ is transmitted over a noisy channel. The received code $R(x) = \gamma_0 + \gamma_1 x + \gamma_2 x^2 + \cdots + \gamma_{N-1} x^{N-1}$ is composed of the original code with the addition of possible errors, i.e.,

$$\gamma(x) = f(x) + e(x)$$

where $e(x) = e_0 + e_1 x + e_2 x^2 + \cdots + e_{N-1} x^{N-1}$ is the error polynomial.

Upon receiving the message $\gamma(x)$, one may decode the message symbols by first using the FFT over $I_{F_n}(\sqrt[8]{2})$. The transform is taken over the received N-tuple message $(\gamma_0, \gamma_1, \cdots, \gamma_{N-1})$, the coefficients of the polynomial $\gamma(x)$. This transform is

$$\begin{aligned} S_K &= \sum_{n=0}^{N-1} \gamma_n (\sqrt[8]{2})^{Kn} \qquad \text{for } K = 0,1, \cdots, N-1 \\ &= \sum_{n=0}^{N-1} (f_n + e_n)(\sqrt[8]{2})^{Kn} \\ &= \sum_{n=0}^{N-1} f_n (\sqrt[8]{2})^{Kn} + \sum_{n=0}^{N-1} e_n (\sqrt[8]{2})^{Kn} \\ &= F_K + E_K \end{aligned}$$

Since $f(x)$ is a multiple of $g(x)$, $f(\alpha^i) = 0$ for $i = 1,2, \cdots, d-1$. Hence,

$$\begin{aligned} S_K = E_K = e((\sqrt[8]{2})^K) &= \sum_{n=0}^{N-1} e_n (\sqrt[8]{2})^{Kn} \\ &= \sum_{n=0}^{N-1} e_n ((\sqrt[8]{2})^n)^K \qquad \text{for } K = 1,2, \cdots, d-1 \end{aligned} \tag{3}$$

Let Y_i and X_i be the ith error magnitude and the ith error location, respectively. Then the syndrome in Eq. (3) becomes

$$S_K = E_K = \sum_{i=1}^{t} Y_i X_i^K \qquad \text{for } K = 1,2, \cdots, d-1 \tag{4}$$

The error locator polynomial $\sigma(x)$ is defined as usual by

$$\sigma(x) = \prod_{i=1}^{t} (1 - X_i x) = 1 - \sigma_1 x + \sigma_2 x^2 + \cdots, (-1)^t \sigma_t x^t$$

where σ_i are the elementary symmetric functions.

It follows that

$$\sigma(X_i^{-1}) = 0 = 1 - \sigma_1 X_i^{-1} + \sigma_2 X_i^{-2} + \cdots, + (-1)^t \sigma_t X_i^{-t}$$
$$\text{for } i = 1,2, \cdots, t$$

Multiplying the above equation by $Y_i X_i^{j+t}$, one gets

$$Y_i X_i^{j+t} - \sigma_1 Y_i X_i^{j+t-1} + \sigma_2 Y_i X_i^{j+t-2} + \cdots, + (-1)^t \sigma_t Y_i X_i^{j+t-t}$$

Summing on i for $i = 1,2, \cdots, t$, then

$$\sum_{i=1}^{t} Y_i X_i^{j+t} - \sigma_1 \sum_{i=1}^{t} Y_i X_i^{j+t-1} + \cdots (-1)^t \sigma_t \sum_{i=1}^{t} Y_i X_i^j = 0$$

Using Eq. (4), we have

$$S_{j+t} - \sigma_1 S_{j+t-1} + \cdots + (-1)^t \sigma_t S_j = 0 \qquad \text{for } j \leq t \tag{5}$$

and

$$E_{j+t} - \sigma_1 E_{j+t-1} + \cdots + (-1)^t \sigma_j E_j = 0 \qquad \text{for } j > t \tag{6}$$

If $S_1 = E_1, S_2 = E_2, \cdots, S_{d-1} = E_{d-1}$ are known, the σ_i for $i = 1, 2, \cdots, t$ in Eq. (5) can be calculated by using Berlekamp's (Ref. 22) iterative algorithm. If σ_i is known, Eq. (6) is then used to obtain $E_0, E_d, E_{d+1}, \cdots, E_{N-1}$, and the transform of the N-tuple error pattern, i.e., $(E_0, E_1, E_2, \cdots, E_{N-1})$ is obtained. Thus, the N-tuple error pattern $(e_0, e_1, \cdots, e_{N-1})$ is found by taking the inverse transform over $I_{F_n}(\sqrt[8]{2})$ of E_K for $K = 0, 1, \cdots, N-1$. Finally, the original N-tuple symbols code can be computed by subtracting e_n from received code γ_n.

To recapitulate, the decoding of systematic Reed-Solomon codes using the transform over $I_{F_n}(\sqrt[8]{2})$, is composed of the following three steps:

(1) Compute the transform over $I_{F_n}(\sqrt[8]{2}) \cong GF(F_n)$ for $n = 3,4$ of the received code N-tuple, i.e.,

$$S_K = \sum_{n=0}^{N-1} \gamma_n \alpha^{nk}$$

where $\gamma_n \epsilon GF(F_n)$ and $\alpha = \sqrt[8]{2} \epsilon GF(F_n)$ is an element of order $N = 2^{n+4}$.

(2) Use Berlekamp's iterative algorithm (Refs. 19 and 22) to determine σ_i from the known $S_j = E_j$ for $i = 1, 2, \cdots, t$ and $j = 1, 2, \cdots, d-1$. Then compute the remaining E_j.

(3) Compute the inverse of the transform over $I_{F_n}(\sqrt[8]{2})$ of $(S_K - E_K)$ to obtain the corrected code.

An advantage of this decoding algorithm over other methods is that a FFT over $GF(F_n)$ can be used to compute the syndromes and error magnitudes. Also the Berlekamp's algorithm can be performed in the arithmetic of $GF(F_n)$. The arithmetic used to perform the *FFT* over $GF(F_n)$ only requires integer additions, circular shifts, and a minimum number of multiplications; thus, such a Fermat number theoretic transform decoder for an RS code of as many as 2^{n+4} symbols can be accomplished faster and simpler than other RS decoders. Since this new transform decoding algorithm is independent of code rate, it is more efficient for correcting a large number of errors in an RS code. The FFT over $GF(F_n)$ becomes more efficient for the longer RS codes.

A disadvantage of this decoding method is that the Fermat primes F_n exist only for $n \leq 4$ and the lengths and dynamic ranges of the transforms associated with these primes are often severly limited. To remedy this it is well known (Ref. 9) that such transforms can be defined over rings of integers modulo a Fermat numbers $F_n = 2^{2^n} + 1$ for $n = 5, 6$, i.e., I_{F_n}. The syndromes can be evaluated by using transforms over this ring. If one knows the S_j, the inverse element in the ring is needed to evaluate the σ_i. However, the inverse of an element in this resulting ring does not exist unless $(a, F_n) = 1$. For this reason, transforms over I_{F_n} cannot be used directly to decode RS codes.

It should be pointed out that a word length of $2^n + 1$ bits is required to represent a number in $GF(F_n)$. However, the word length in the transmitted word is often a multiple 4 bits. Thus, the values of the symbols in $GF(F_n)$ cannot be represented easily as a 2^n bit word. To remedy this, suppose the information symbols are represented in the range from 0 to $2^{2^n} - 1$. After encoding the information symbols, the parity check symbols may occur in the range between 0 and 2^{2^n}. If 2^{2^n} is observed as a parity check symbol, deliberately change this value to 0, now an error. The transform decoder will correct this error automatically.

A simple example of the above decoding procedure for an RS code in $GF(F_n)$ is now presented.

Example. Let $GF(2^{2^2} + 1)$ be the field of integers modulo the Fermat prime $F_2 = 17$. We consider a 2-error correcting 8-tuple RS code in $GF(17)$.

Since $2^{2^2} \equiv -1 \bmod 17$, by (Ref. 12, theorem 1) $\alpha = 2$ is an element of order 8 in $GF(17)$. The cyclic subgroup of 8 elements with generator $\alpha = 2$ in GF (17) follows:

$$\alpha = 2$$

$$\alpha^2 = 2^2 = 4$$

$$\alpha^3 = 2^3 = 8$$

$$\alpha^4 = 2^4 = -1$$

$$\alpha^5 = 2^5 = 15$$

$$\alpha^6 = 2^6 = 13$$

$$\alpha^7 = 2^7 = 9$$

$$\alpha^8 = 2^8 = 1$$

The block length of the RS code is $N = 8$. It can correct at most $t = 2$ errors. This implies that the minimum distance of the code is $d = 2t + 1 = 5$. Then the information symbols are $N - (d - 1) = 8 - (5 - 1) = 4$. The generator polynomial is defined as

$$g(x) = \sum_{i=1}^{d-1} (x - \alpha^i) = \sum_{i=1}^{5-1} (x - 2^i)$$
$$= x^4 + 4x^3 + 8x^2 + 9x + 4$$

Assume the information symbols are $1, 2, 3, 2 \epsilon GF(17)$, i.e., $I(x) = 1x^7 + 2x^6 + 3x^5 + 2x^4$. Recall that the code word is a multiple of $g(x)$. By the division algorithm, one gets

$$I(x) = q(x)g(x) + R(x)$$

where $R(x)$ is the remainder of polynomial of degree less than the degree of $g(x)$. It follows that

$$f(x) = I(x) - R(x) = q(x)g(x)$$

Hence the encoding of $I(x)$ is the polynomial

$$f(x) = 5 + 2x + 12x^2 + 15x^3 + 2x^4 + 3x^5 + 2x^6 + x^7$$
$$= (5, 2, 12, 15, 2, 3, 2, 1)$$

Suppose that two errors occur in the received word, e.g.,

$$\gamma(x) = 5 + 2x + 9x^2 + 15x^3 + 2x^4 + 1x^5 + 2x^6 + 1x^7$$
$$= (\gamma_0, \gamma_1, \gamma_2, \gamma_3, \gamma_4, \gamma_5, \gamma_6, \gamma_7)$$
$$= (5, 2, 12 - 3, 15, 2, 3 - 2, 2, 1)$$

Then the error pattern $e(x)$ is

$$e(x) = \gamma(x) - f(x)$$
$$= 0 + 0 \cdot x^1 - 3x^2 + 0 \cdot x^3 + 0 \cdot x^4 - 2x^5 + 0 \cdot x^6 + 0 \cdot x^7$$
$$= (e_0, e_1, e_2, e_3, e_4, e_5, e_6, e_7)$$
$$= (0, 0, 14, 0, 0, 15, 0, 0)$$

The syndrome can be calculated, using an FFT over $GF(F_n)$. That is,

$$S_K = E_K = \sum_{n=0}^{8-1} \gamma_n 2^{nK}$$
$$= \sum_{n=0}^{8-1} (f_n + e_n) 2^{nK}$$
$$= \sum_{n=0}^{8-1} f_n 2^{nK} + \sum_{n=0}^{8-1} e_n 2^{nK}$$
$$= F_K + E_K$$

Since

$$\sum_{n=0}^{8-1} f_n 2^{nK} = 0 \qquad \text{for } K = 1, 2, 3, 4$$

then

$$S_K = E_K = \gamma(x) = e(x) = \sum_{n=0}^{8-1} e_n \cdot 2^{nK} = \sum_{i=1}^{2} Y_i X_i^K$$

$$\text{for } K = 1, 2, 3, 4$$

Hence,

$$S_1 = e(2) = E_1 = -3 \cdot 2^2 - 2 \cdot 2^5 = -8$$
$$S_2 = e(2^2) = E_2 = -3(2^2)^2 - 2(2^2)^5 = -5$$
$$S_3 = e(2^3) = E_3 = -3(2^3)^2 - 2(2^3)^5 = 11$$
$$S_4 = e(2^4) = E_4 = -3(2^4)^2 - 2(2^4)^5 = -1$$

The error locator polynomial $\sigma(x)$ is

$$\sigma(x) = \prod_{i=1}^{t} (1 - X_i x) = \prod_{i=1}^{2} (1 - X_i x)$$
$$= 1 - (X_1 + X_2)x + X_1 X_2 x^2$$
$$= 1 - \sigma_1 x + \sigma_2 x^2$$

where

$$\sigma_1 = X_1 + X_2, \sigma_2 = X_1 X_2$$

It follows that

$$\sigma(X_i^{-1}) = 0 = 1 - \sigma_1 X_i^{-1} + \sigma_2 X_i^{-2} \qquad \text{for } i = 1, 2$$

Multiplying by $Y_i X_i^{j+1}$,

$$Y_i X_i^{j+2} - \sigma_1 Y_i X_i^{j+2-1} + \sigma_2 Y_i X_i^{j+2-2} = 0 \qquad \text{for } i = 1, 2$$

Summing on i for $i = 1, 2$, one has

$$\sum_{i=1}^{2} Y_i X_i^{j+2} - \sigma_1 \sum_{i=1}^{2} Y_i X_i^{j+1} + \sigma_2 \sum_{i=1}^{2} Y_i X_i^{j} = 0$$

i.e.,

$$S_{j+2} - \sigma_1 S_{j+1} + \sigma_2 S_j = 0, \qquad \text{for } j \leq 2 \tag{7}$$

and

$$E_{j+2} - \sigma_1 E_{j+1} + \sigma_2 E_j = 0 \qquad \text{for } j > 2 \tag{8}$$

It follows from Eq. (7) that for $j = 1, 2$

$$S_3 - \sigma_1 S_2 + \sigma_2 S_1 = 0$$

$$S_4 - \sigma_1 S_3 + \sigma_2 S_2 = 0$$

or

$$-5\sigma_1 + \sigma_2 8 \equiv 11 \bmod 17 \tag{9a}$$

$$11\sigma_1 + \sigma_2 5 \equiv 1 \bmod 17 \tag{9b}$$

Since

$$\begin{vmatrix} -5, 8 \\ 11, 5 \end{vmatrix} = -25 - 11 \cdot 8 \equiv 6 \not\equiv 0 \bmod 17$$

Equation (9) has a solution. To obtain the solution, multiply Eq. (9a) by 11 and Eq. (9b) by 5, then

$$-55\sigma_1 + \sigma_2 88 = 36$$

$$55\sigma_1 + \sigma_2 25 = -5$$

The solutions of above equations are

$$\sigma_2 = \frac{-3}{11} \equiv -3 \cdot 11^{-1} \equiv -3 \cdot 14 \equiv 9 \bmod 17$$

and

$$\sigma_1 = \frac{-46}{11} \equiv 5 \cdot (-3) \equiv 2 \bmod 17$$

Equation (8) becomes

$$E_{j+2} - 2E_{j+1} + 9E_j = 0 \qquad \text{for } j > 2 \tag{10}$$

From Eq. (10), one gets the rest of the transform E_j of the error pattern, i.e.,

$$E_5 = 2E_4 - 9E_3 \equiv 2(-1) - 9(11) \equiv 1 \bmod 17$$

$$E_6 = 2E_5 - 9E_4 \equiv 2(1) - 9(-1) \equiv 11 \bmod 17$$

$$E_7 = 2E_6 - 9E_5 \equiv 2(11) - 9(1) \equiv 13 \bmod 17$$

$$E_8 = 2E_7 - 9E_6 \equiv 2(13) - 9(11) \equiv 12 \bmod 17$$

The inverse transform over $GF(2^2 + 1)$ of the E_j is

$$e_n = (8)^{-1} \sum_{K=0}^{8-1} E_K 2^{-nK} \qquad \text{for } 0 \leq n \leq 7$$

$$= (-2)(E_0 2^0 + E_1 2^{-n} + E_2 2^{-2n} + E_3 2^{-3n} + E_4 2^{-4n} + E_5 2^{-5n} + E_6 2^{-6n} + E_7 2^{-7n})$$

$$= (-2)(12 \cdot 2^0 + 9 \cdot 2^{-n} + 12 \cdot 2^{-2n} + 11 \cdot 2^{-3n} + 16 \cdot 2^{-4n} + 1 \cdot 2^{-5n} + 11 \cdot 2^{-6n} + 13 \cdot 2^{-7n})$$

Using the FFT algorithm, we have finally

$$(e_0, e_1, e_2, e_3, e_4, e_5, e_6, e_7) = (0, 0, 14, 0, 0, 15, 0, 0)$$

The corrected codeword is

$$f(x) = R(x) - e(x)$$

$$= (5, 2, 9, 15, 2, 1, 2, 1) - (0, 0, 14, 0, 0, 15, 0, 0)$$

$$= (5, 2, 12, 15, 2, 3, 2, 1)$$

IV. A Fast Transform Over $GF(F_n)$ for the Nonsystematic Reed-Solomon Codes

The transform over finite field $GF(q)$, where q is a prime, can be also used to decode nonsystematic Reed-Solomon codes. The nonsystematic Reed-Solomon code is defined in Ref. 5. Let $\alpha \epsilon GF(q)$ be an element of order N. Consider the information polynomial $I(x)$ with coefficients $\epsilon GF(q)$, i.e.,

$$I(x) = i_0 + i_1 x + \cdots, + i_{K-1} x^{K-1}$$

The transmitted code word is the following polynomial:

$$f(x) = I(\alpha^0) + I(\alpha^1)x + I(\alpha^2)x^2 + \cdots, + I(\alpha^{N-1})x^{N-1}$$

$$= F_0 + F_1 x + F_2 x^2 + \cdots, + F_{N-1} x^{N-1}$$

where the $I(\alpha^i)$ are obtained by using the transform over $GF(F_n)$. That is,

$$I(\alpha^K) = F_K = \sum_{n=0}^{N-1} i_n (\alpha^K)^n \qquad \text{for } K = 0, 1, 2, \cdots, N-1$$

where

$$i_0 = i_0, i_1 = i_1, \cdots, i_{K-l} = i_{K-l}, i_K = 0, i_{K+1} = 0, \cdots, i_{N-1} = 0$$

The inverse transform over $GF(q)$ of F_K is

$$i_n = (N)^{-1}\sum_{K=0}^{N-1} F_K\alpha^{-nK} \qquad \text{for } n = 0, 1, \cdots, N-1$$
$$= (N)^{-1}f(\alpha^{-n})$$

It follows that

$$(N)^{-1}f(\alpha^{-n}) = i_n \qquad \text{for } n = 0, 1, \cdots, K-1$$
$$= 0 \qquad \text{for } n = K, \cdots, N-1$$

Suppose the received word in the transform domain is

$$\gamma(x) = f(x) + e(x)$$
$$= f(x) + \sum_{j=0}^{N-1} e_j\alpha^{jn}$$

where $e(x)$ is the transform of the error pattern. Then, the inverse transform of $\gamma(x)$ is

$$S_n = (N)^{-1}\sum_{K=0}^{N-1}\gamma_k\alpha^{-Kn}$$
$$= (N)^{-1}\sum_{K=0}^{N-1} -f_k\alpha^{-Kn} + (N)^{-1}\sum_{K=0}^{N-1}\left(\sum_{j=0}^{N-1} e_j\alpha^{jn}\right)\alpha^{-Kn}$$
$$= (N)^{-1}f(\alpha^{-n}) + e_n$$

or

$$S_n = i_n + e_n \qquad \text{for } n = 0, 1, \cdots, K-1$$
$$= 0 + e_n \qquad \text{for } n = K, \cdots, N-1$$

Thus, the syndrome is

$$S_n = e_n \qquad \text{for } n = K, K+1, \cdots, N-1 \qquad (11)$$

Hence $t = (N-K)/2$ errors in N-tuple can be corrected in the nonsystematic RS code. By the same procedure, used in the derivation of Eq. (5), one gets

$$S_{N-j} - \sigma_1 S_{N-j-1} + \cdots + \sigma_t(-1)^t S_{N-j-t} = 0 \qquad (12)$$
$$\text{for } 1 \le j \le t$$

and

$$e_{N-j} - \sigma 1 e_{N-j-1} + \cdots + (-1)^t\sigma_t e_{N-j-t} = 0 \qquad (13)$$
$$\text{for } j \ge t$$

Using the Berlekamp's algorithm, σ_i can be computed for the syndromes. The error pattern can be obtained by using Eq. (13). We see that a fast transform over $GF(q)$ for the nonsystematic RS code can be implemented by using only one inverse transform. However, encoding is accomplished by a forward transform. Hence for the nonsystematic RS codes the FFT over $GF(F_n)$ is both to encode and to decode the codes.

V. A Transform Decoder Over the Finite Field, $GF(K \cdot 2^n + 1)$

In the previous section, transforms over the field of type $I_{F_n}(\sqrt[8]{2})$ were defined to decode RS code. However, Fermat primes existed only for $n \le 4$, and the lengths and dynamic ranges of the transforms associated with these primes were often severely limited. Also it was found that a word length, 4m could not always be represented adequately in $GF(F_n)$. To alleviate such difficulties another approach was proposed recently (Refs. 16 and 17). High-speed number theoretic transforms were defined on the Galois field $GF(q)$, where the prime q was of form $q = K \times 2^n + 1$, where n and K are integers.

In Ref. 16, an FFT algorithm over $GF(q)$ was utilized to realize transforms of integers. Such transforms offer a substantial variety of possible transform lengths and dynamic ranges. However, the arithmetic needed was often somewhat more extensive than required for the Fermat primes.

If q is a prime of the form $K \cdot 2^n + 1$, by Eqs. (2a) and (2b), a transform can be defined on $GF(K \times 2^n + 1)$. The order of the multiplicative group with generator of $GF(q)$ is given by

$$t = q - 1 = K \cdot 2^n$$

Since t has the factor 2^n the usual radix-2 FFT algorithm can be utilized to calculate the transform of as many as $d = 2^n$ points. If $d = 2^m$, $1 \le m \le n$ and α is the primitive element of $GF(q)$, then the generator of G_d is evidently $\gamma = \alpha^{2^{n-m}}$. Primes of the form $K \cdot 2^n + 1$ can be found in the table of Ref. 23. Thus primes of the form $K \cdot 2^n + 1$ can be chosen to fit into the word lengths of different digital computers.

To fit a transform defined by Eq. (2a) in the PDP-10 computer, which has a word length of 36 bits, the largest prime number of the form $K \times 2^n + 1$ was found to be the prime $45 \times 2^{29} + 1$. By Fermat's theorem, $2^{45 \cdot 2^{29}} \equiv 1 \bmod q$, where $q = 45 \times 2^{29} + 1$. This is equivalent to $(2^{45})^{2^{29}} \equiv 1 \bmod q$. It can be verified by a computer program that $(2^{45})^{2^{27}} \equiv -1 \bmod q$. Thus, by theorem 1 of

Ref. 12, $2^{45} \equiv 8589933136 \bmod q$ is an element of order 2^{28}, where $q = 45 \times 2^{29} + 1$. It follows that $\gamma \equiv 2^{45 \cdot 2^{28-K}} \bmod q$ is an element of order 2^K where $0 \leq K \leq 28$. A detailed discussion for finding the index or order of an element modulo a prime of form $K \cdot 2^n + 1$ can be found in Ref. 16.

Multiplication modulo of the prime number $q = 45 \times 2^{29} + 1$ is straightforwardly performed in assembly language software in the PDP-10 computer. To perform addition modulo q, let $A + C = A + (C - q)$, where $(C - q) \leq 0$. Then if $A + (C - q) \leq 0$, the addition is accomplished by the add command, otherwise it equals $A + (C - q) + q$. Another method for performing addition modulo $K \cdot 2^n + 1$ was developed for small K in Ref. 16.

Subtraction modulo q, if $A - C \leq 0$, is accomplished by the subtract command; otherwise, it equals $A - C + q$. For a more detailed discussion for implementing the transform over $GF(45 \times 2^{29} + 1)$ in software, see Ref. 17. It was shown (Ref. 17) that the arithmetic used to perform this transform requires $d \log d$ integer multiplications mod q and $d \log d$ integer additions mod q. Hence, using the same procedure described in the previous section, a transform over $GF(q)$ where $q = K \cdot 2^n + 1$ can be used to decode a very long RS code with improved symbol range.

VI. A Transform Decoder Over $GF(q^2)$ Where q is a Mersenne Prime

In the previous sections, transforms were defined in $GF(K \times 2^n + 1)$. In this section, a transform is defined on $GF(q^2)$ where q is a Mersenne Prime. It will be shown that the radix-8 FFT over $GF(q^2)$ can be used to decode very long RS code with a goodly number of symbols. Of special interest is the radix-8 FFT algorithm over $GF(q^2)$.

In Ref. 12, Reed and Truong extended previous transforms of Rader (Ref. 7) by developing a Fourier-like transform over the Galois field $GF(q^2)$, a finite field of q^2 elements, where q is a prime, of the form

$$A_K = \sum_{n=0}^{d-1} a_n \gamma^{kn} \qquad \text{for } 0 \leq K \leq d-1 \qquad (14a)$$

In Eq. (14a) the transform length, d, divides $q^2 - 1, a_n \epsilon GF(q^2)$ and γ is a primitive dth root of unity that generates the d-element cyclic subgroup

$$G_d = \{\gamma, \gamma^2, \cdots, \gamma^{d-1}, 1\}$$

in the multiplicative subgroup of $GF(q^2)$. The inverse transform of Eq. (14a) is

$$a_m = (d)^{-1} \sum_{k=0}^{d-1} A_K \gamma^{-Km} \qquad \text{for } 0 \leq m \leq d-1 \quad (14b)$$

where $(d)^{-1}$ denotes the multiplicative inverse of the residue of d modulo q in $GF(q^2)$.

It is shown in Eq. 12 that if q is a Mersenne prime of the form

$$q = 2^p - 1 \qquad \text{for } p = 2, 3, 5, 7, 13, 17, 19, 31, 61, 127, \cdots$$

then the polynomial

$$p(x) = x^2 + 1$$

is always irreducible in $GF(q)$, a finite field of q elements. Since every irreducible quadratic polynomial over $GF(q)$ must split over $GF(q^2)$ (Ref. 24), the existence of a root $\hat{i}$ of the polynomial

$$p(x) = x^2 + 1 = 0$$

is guaranteed in the extension field $GF(q^2)$. Hence $GF(q^2)$ can be constructed as the set

$$GF(q^2) = \{a + \hat{i}b \mid a, b \epsilon GF(q)\}$$

Furthermore, since the mapping from the complex integer field C composed of the set

$$C = \left\{\alpha + \hat{i}\beta \mid \alpha, \beta \text{ integers and } -\frac{q-1}{2} \leq \alpha, \beta \leq \frac{q-1}{2}\right\}$$

where $\hat{i} = \sqrt{-1}$, to $GF(q^2)$ is one-to-one and onto, circular convolutions of complex integers can be performed either in C or in $GF(q^2)$. It is also shown in Ref. 12 that FFTs of as many as 2^{p+1} points can be carried out in $GF(q^2)$.

It was shown (Ref. 14) that the arithmetic operations for performing the transform pair, Eqs. (14a) and (14b), in $GF(q^2)$ requires only modulo q additions, modulo q multiplications, circular shifts of a p-bit register, and complement operations. Also because of the symmetry properties of the dth roots of unity in $GF(q^2)$, where d divides 2^{p+1}, algorithms analogous to the conventional twiddle factor FFT algorithms can be used to compute transforms over $GF(q^2)$.

It was shown (Ref. 14) that when d divides 2^{p+1}, the components of $\gamma^{kd/8}$, where γ is a primitive dth root of unity in $GF(q^2)$ and k is an odd integer, are fixed powers of 2. As a consequence, complex multiplications involving $\gamma^{kd/8}$ can be accomplished merely by additions and circular shifts of $(p-1)/2$ bits in a p-bit register. Therefore, these new FFT algorithms can be made faster and simpler than the conventional FFT algorithm. Of particular interest is a new radix-8 FFT algorithm that requires no multiplications at all when evaluating the set of 8-point discrete Fourier transforms (DFTs) before referencing with the twiddle factor. Hence, using a procedure similar to that discussed in Section 3, a radix-8 FFT over $GF(q^2)$ can be developed to decode RS code of as many as 2^{p+1} symbols.

Observe that the element $c = a + \hat{i}b$ in $GF(q^2)$ can be used to represent two symbols a, b in $GF(q)$. Thus, the transform over $GF(q^2)$ can be used to decode a 2^{p+2}-tuple in $GF(q)$ RS code. By theorem 1 of Ref. 14, we know that an 8th root of unity in $GF(q^2)$ is $\pm 2^{(p-1)/2}(1+\hat{i})$. If special interconnections are made between the inputs and outputs of the p-bit register, the $(p-1)/2$-bit circular shift could be performed readily in one clock time. This fact makes possible an 8-point FFT over $GF(q^2)$, which requires only circular shifts and additions. This FFT over $GF(q^2)$ can be used to decode a special 4-error correcting 16-tuple RS code faster and simpler than any other code of comparable error correcting capability.

The flow chart (Fig. 1) illustrates transform decoding over $GF(q^2)$ for correcting this RS code.

Example: Let $GF(q^2) = \{a + \hat{i}b \mid a, b \epsilon GF(q)\}$ be a Galois field, where $q = 2^3 - 1 = 7$. The information symbols are the 4-tuple $(1+\hat{i},\ 1+2\hat{i},\ 2+\hat{i},\ 1+\hat{i}0)$. Encode this 4-tuple into an 8-tuple by using the FFT over $GF(q^2)$. This code corrects at most two-symbol errors.

From Ref. 14, $\alpha = 2^{(p-1)/2}(1+\hat{i}) = 2^{(3-1)/2}(1+\hat{i}) = 2(1+\hat{i})$ is an element of order 8. That is,

$$\alpha = 2(1+\hat{i})$$
$$\alpha^2 = 2^2(1+\hat{i})^2 = \hat{i}$$
$$\alpha^3 = \hat{i}2(1+\hat{i}) = 2(-1+\hat{i})$$
$$\alpha^4 = -1$$
$$\alpha^5 = 2(-1-\hat{i})$$
$$\alpha^6 = -\hat{i}$$
$$\alpha^7 = -\hat{i}2(1+\hat{i}) = 2(1-\hat{i})$$

The information polynomial is

$$\begin{aligned} E(x) &= i_0 + i_1x + i_2x^2 + i_3x^3 + i_4x^4 + i_5x^5 + i_6x^6 + i_7x^7 \\ &= ((1+\hat{i}), (1+2\hat{i}), (2+\hat{i}), (1+\hat{i}0), (0+\hat{i}0), \\ &\quad (0+\hat{i}0), (0+\hat{i}0), (0+\hat{i}0)) \end{aligned}$$

Encoding is performed by taking the FFT over $GF(7^2)$ of i_n That is,

$$\begin{aligned} F_K &= \sum_{n=0}^{8-1} i_n \alpha^{nK} \\ &= (1+\hat{i}) + (1+2\hat{i})\alpha^K + (2+\hat{i})\alpha^{2K} + (1+\hat{i}0)\alpha^{3K} \end{aligned}$$

this implies

$$\begin{aligned} f(x) &= F_0 + F_1x + F_2x^2 + F_3x^3 + F_4x^4 + F_5x^5 + F_6x^6 + F_7x^7 \\ &= ((5+4\hat{i}), (3+4\hat{i}), 4, (-2-\hat{i}), 1, (4+2\hat{i}), 1, (-\hat{i}-1)) \end{aligned}$$

Suppose the received word associated with two errors is

$$\begin{aligned} \gamma(x) &= \gamma_0 + \gamma_1x + \gamma_2x^2 + \gamma_3x^3 + \gamma_4x^4 + \gamma_5x^5 + \gamma_6x^6 + \gamma_7x^7 \\ &= ((1+\hat{i}), (3+4\hat{i}), 4, (-2-\hat{i}), 1, (2+\hat{i}), 1, (-\hat{i}-1)) \\ &= f_n + e_n \\ &= ((5+4\hat{i}), (3+4\hat{i}), 4, (-2-\hat{i}), 1, (4+2\hat{i}), 1, (-\hat{i}-1)) \\ &\quad + ((-4-3\hat{i}), 0, 0, 0, 0, (-2-\hat{i}), 0, 0) \end{aligned}$$

The inverse transform over $GF(q^2)$ of γ_n is defined as

$$S_n = 2^{-3}\sum_{K=0}^{8-1} \gamma_k \alpha^{-nK} = \sum_{K=0}^{8-1} \gamma_K \alpha^{-nK}$$

Using the FFT algorithm, one gets,

$$S_0 = 2 + 4\hat{i}$$
$$S_1 = -4 - 3\hat{i}$$
$$S_2 = -3$$
$$S_3 = 2 + 5\hat{i}$$
$$S_4 = e_4 = -2 - 2\hat{i}$$
$$S_5 = e_5 = -3 - \hat{i}$$
$$S_6 = e_6 = -3 + 2\hat{i}$$
$$S_7 = e_7 = -2 + 3\hat{i}$$

Substituting S_i for $i = 4, 5, 6, 7$ in Eq. (12), yields

$$\left.\begin{aligned} \sigma_1 S_5 - \sigma_2 S_4 &= S_6 \\ \sigma_1 S_6 - \sigma_2 S_5 &= S_7 \end{aligned}\right\} \qquad (15)$$

Since

$$\begin{vmatrix} S_5 & -S_4 \\ S_6 & -S_5 \end{vmatrix} = -S_5^2 + S_4 S_6$$
$$= -(-3 - \hat{i})^2 + (-2 - 2\hat{i})(-3 + 2\hat{i})$$
$$= 2 + 3\hat{i} \neq 0$$

Eq. (15) has a solution. Thus

$$\sigma_1 = \frac{-S_6 S_5 + S_4 S_7}{-S_5^2 + S_4 S_6}$$
$$= \frac{-(-3 + 2\hat{i})(-3 - \hat{i}) + (-2 - 2\hat{i})(-2 + 3\hat{i})}{2 + 3\hat{i}}$$
$$= \frac{-1 + \hat{i}}{2 + 3\hat{i}}$$
$$\sigma_2 = \frac{S_5 S_7 - S_6^2}{-S_5^2 - S_4 S_6} = \frac{-3 - 2\hat{i}}{2 + 3\hat{i}}$$

To find the inverse element $(a + \hat{i}b)$ of $(2 + 3\hat{i})$ in $GF(7^2)$

$$(2 + 3\hat{i})(a + \hat{i}b) \equiv 1 \bmod 7$$

which implies

$$\begin{aligned} 2a - 3b &\equiv 1 \bmod 7 \\ 2b + 3a &\equiv 0 \bmod 7 \end{aligned} \qquad (16)$$

The solutions of Eq. (16) are

$$a = -2$$
$$b = 3$$

Thus,

$$(2 + 3\hat{i})^{-1} = (-2 + \hat{i}3)$$

Hence

$$\sigma_1 = (-1\hat{i})(2 + 3\hat{i})^{-1} = -1 + 2\hat{i}$$
$$\sigma_2 = (-3 - 2\hat{i})(2 + 3\hat{i})^{-1} = 5 + 2\hat{i}$$

From Eq. (13),

$$e_{j+2} = (-1 + 2\hat{i})e_{j+1} - (5 + 2\hat{i})e_j \qquad \text{for } j = 6, 7, 8, 9$$

It follows that

$$e_0 = \sigma_1 e_7 - \sigma_2 e_6$$
$$= (-1 + 2\hat{i})(-2 + 3\hat{i}) - (5 + 2\hat{i})(-3 + 2\hat{i})$$
$$= 1 + 3\hat{i}$$
$$e_1 = \sigma_1 e_0 - \sigma_2 e_7$$
$$= (-1 + 2\hat{i})(1 + 3\hat{i}) - (-2 + 2\hat{i})(-2 + 3\hat{i})$$
$$= 2 + 2\hat{i}$$
$$e_2 = \sigma_1 e_1 - \sigma_2 e_0$$
$$= (-1 + 2\hat{i})(2 + 2\hat{i}) - (-2 + 2\hat{i})(1 + 3\hat{i})$$
$$= 2 - 2\hat{i}$$
$$e_3 = \sigma_1 e_2 - \sigma_2 e_1$$
$$= (-1 + 2\hat{i})(2 - \hat{i}) - (-2 + 2\hat{i})(2 + 2\hat{i})$$
$$= -2\hat{i} + 1$$

Hence the error pattern is

$$e_0 = 1 + 3\hat{i}$$
$$e_1 = 2 + 2\hat{i}$$
$$e_2 = 2 - \hat{i}$$
$$e_3 = 1 - 2\hat{i}$$
$$e_4 = -2 - 2\hat{i}$$
$$e_5 = -3 - \hat{i}$$
$$e_6 = -3 + 2\hat{i}$$
$$e_7 = -2 + 3\hat{i}$$

Since $S_n = i_n + e_n$, the corrected code is

$$i_n = S_n - e_n$$
$$= (2 + 4\hat{i}, -4 - 3\hat{i}, -3, 2 + 5\hat{i}, -2 - 2\hat{i},$$
$$-3 - \hat{i}, -3 + 2\hat{i}, -2 + 3\hat{i}) - (1 + 3\hat{i}, 2$$
$$+ 2\hat{i}, 2 - \hat{i}, -2\hat{i} + 1, -2 - 2\hat{i}, -3 - \hat{i},$$
$$-3 + 2\hat{i}, -2 + 3\hat{i})$$
$$= (1 + \hat{i}, 1 + 2\hat{i}, 2 + \hat{i}, 1 + 0\hat{i}, 0 + \hat{i}0, 0 + \hat{i}0,$$
$$0 + \hat{i}0, 0 + \hat{i}0)$$

Appendix

A Method for Determining Octadic Residue and Octadic Nonresidues of a Prime

To define the field of type $I_{F_n}(\sqrt[8]{2})$, it is necessary to determine whether 2 is either an octadic residue or an octadic nonresidue of F_n. Towards this end, the following definition and theorems are needed.

Definition. Suppose m is an integer and q is a prime such that $(m, q) = 1$. Let $[m/q]_2$ be the Gauss-Legendre symbol. Then $[m/q]_{2^n}$ is the symbol, defined by

$$\left[\frac{m}{q}\right]_{2^n} = +1 \quad \text{if } X^{2^n} \equiv m \bmod q \text{ has an integer solution in } GF(q)$$

$$= -1 \quad \text{if } X^{2^n} \equiv m \bmod q \text{ has not an integer solution in } GF(q) \text{ for which } [m/q]_{2^{n-1}} = 1$$

Theorem 1. Let $q = a^2 + b^2 = 4n + 1$ be a prime for a odd and b even. If $[2/q]_2 = 1$, then $[2/q]_4 = (-1)^{b/4}$.

Theorem 2. Let $q = a^2 + b^2 = 8n + 1$ be a prime, a odd and b even. If $[2/q]_4 = 1$, then

$$\left[\frac{2}{q}\right]_8 = (-1)^{b/8} \quad \text{if } n \text{ is even}$$

$$= (-1)^{(b/8)+1} \quad \text{if } n \text{ is odd}$$

For the proof of these two theorems, see Ref. 20.

Let q be a Fermat prime, i.e., $F_n = 2^{2^n} + 1 = 4 \cdot 2^{2^n-2} + 1 = (2^{2^{n-1}})^2 + 1 = a^2 + b^2$ for $n = 1, 2, 3, 4$.

By Theorem 1,

$$\left[\frac{2}{F_n}\right]_4 = (-1)^{2^{2^n-1}/4} = +1 \text{ for } n = 3, 4$$

Also by theorem 2,

$$\left[\frac{2}{F_n}\right]_8 = (-1)^{2^{2^n-1}/8} = 1 \quad \text{for } n = 3, 4$$

Thus, 2 is an octadic residue modulo F_n for $n = 3, 4$.

Acknowledgment

The authors wish to thank Mr. B. Mulhall and Dr. B. Benjauthrit of JPL for their early support, suggestions, and encouragement of the research that led to this paper.

References

1. Gove, W. C., *Transmitting Binary Symbols with Reed-Solomon Code.* Johns Hopkins EE Report No. 73-5, April 1973.

2. Mandelbaum, D., "On Decoding Reed-Solomon Codes," *IEEE Transactions on Information Theory*, Vol. IT-17, No. 6, pp. 707–712, November 1971.

3. Reed, I. S., and Solomon, G., "Polynomial Codes over Certain Finite Fields," *SIAM J. Appl. Math.*, Vol. 8, pp. 300–304, June 1960.

4. Michelson, A., *A New Decoder for the Reed-Solomon Codes Using a Fast Transform Technique.* Systems Engineering Technical Memorandum No. 52, Electronic Systems Group Eastern Division GTE Sylvania, August 1975.

5. Peterson, W. W., *Error-Correcting Codes.* MIT Press, Cambridge, Mass., 1961, pp. 168–169.

6. Achonhage, S., and Strassen, V., "Schnelle Multiplikation Grosser Zahlem," *Computing 7*, pp. 281–292, 1971.

7. Rader, C. M., "Discrete Convolution via Mersenne Transforms," *IEEE Trans. on Computers*, Vol. C-21, No. 12, December 1972.

8. Agarwal, R. C., and Burrus, C. S., "Fast Convolution Using Fermat Number Transforms with Applications to Digital Filtering," *IEEE Trans. on Acoustics. Speed, and Signal Processing*, Vol. ASSP-22, No. 2, April 1974.

9. Agarwal, R. C., and Burrus, C. S., "Number Theoretic Transforms to Implement Fast Digital Convolutional," *Proceedings of the IEEE*, Vol. 63, No. 4, April 1975.

10. Reed, I. S., and Truong, T. K., "Convolution Over Residue Classes of Quadratic Integers," in *IEEE Trans. Inf. Theory*, July 1976.

11. McClellan, J. H., "Hardware Realization of a Fermat Number Transform," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, Vol. Assp. 24, No. 3, June 1976.

12. Reed, I. S., and Truong, T. K., "The Use of Finite Fields to Compute Convolutions," *IEEE Trans. Inf. Theory*, Vol. IT-21, No. 2, pp. 208–212, March 1975.

13. Reed, I. S., *The Use of Finite Fields and Rings to Compute Convolutions,* Laboratory Technical Memorandum No. 24L-0012, MIT Lincoln Laboratory, October 1973.

14. Liu, K. Y., Reed, I. S., and Truong, T. K., "Fast Algorithms for Complex Integer Transforms," submitted to *IEEE Trans. on Acoustics, Speed, and Signal Processing.*

15. Reed, I. S., and Truong, T. K., "Complex Integer Convolutions over a Direct Sum of Galois Fields," *IEEE Trans. Inform. Theory*, Vol. IT-21, November 1975.

16. Golomb, S. W., Reed, I. S., and Truong, T. K., "Integer Convolutions Over the Finite iFeld *GF* $(3 \cdot 2^n + 1)$," to be published in *SIAM Journal on Applied Mathematics.*

17. Reed, I. S., Truong, T. K., Kwoh, Y. S., and Hall, E. L., "Image Processing by Transforms over a Finite Field," submitted to *IEEE Transactions on Computers*, January 1976. Available from the Computer Society Repository.

18. Singleton, R. C., "An Algorithm for Computing the Mixed Radix Fast Fourier Transform," *IEEE Trans. Audio Electroacoust.*, Vol. AU-17, pp. 93–103, June 1969.

19. Justesen, J., "On the Complexity of Decoding of Reed-Solomon Codes," *IEEE Trans. Inform. Theory*, Vol IT-22, March 1976.

20. Ping-Yuan Wu, *A Rational Reciprocity Law*, Ph.D. dissertation, Dept. of Mathematics, University of Southern California, August 1975.

21. Reed, I. S., and Truong, T. K., "Convolutions over Quartic Integer Residue Classes," submitted to *IEEE Trans. Inf. Theory.*

22. Berlekamp, E. R., *Algebraic Coding Theory*, New York, McGraw Hill, 1968, Chapter 7.

23. Robinson, R. M., "A Report on Primes of the Form $K \cdot 2^n + 1$ and on Factors of Fermat Numbers," *Proceedings of the American Mathematical Society*, Vol. 9, No. 5, October 1958.

24. Herstein, I. N., *Topics in Algebra*, Blaisdell Publishing Co., 1964.

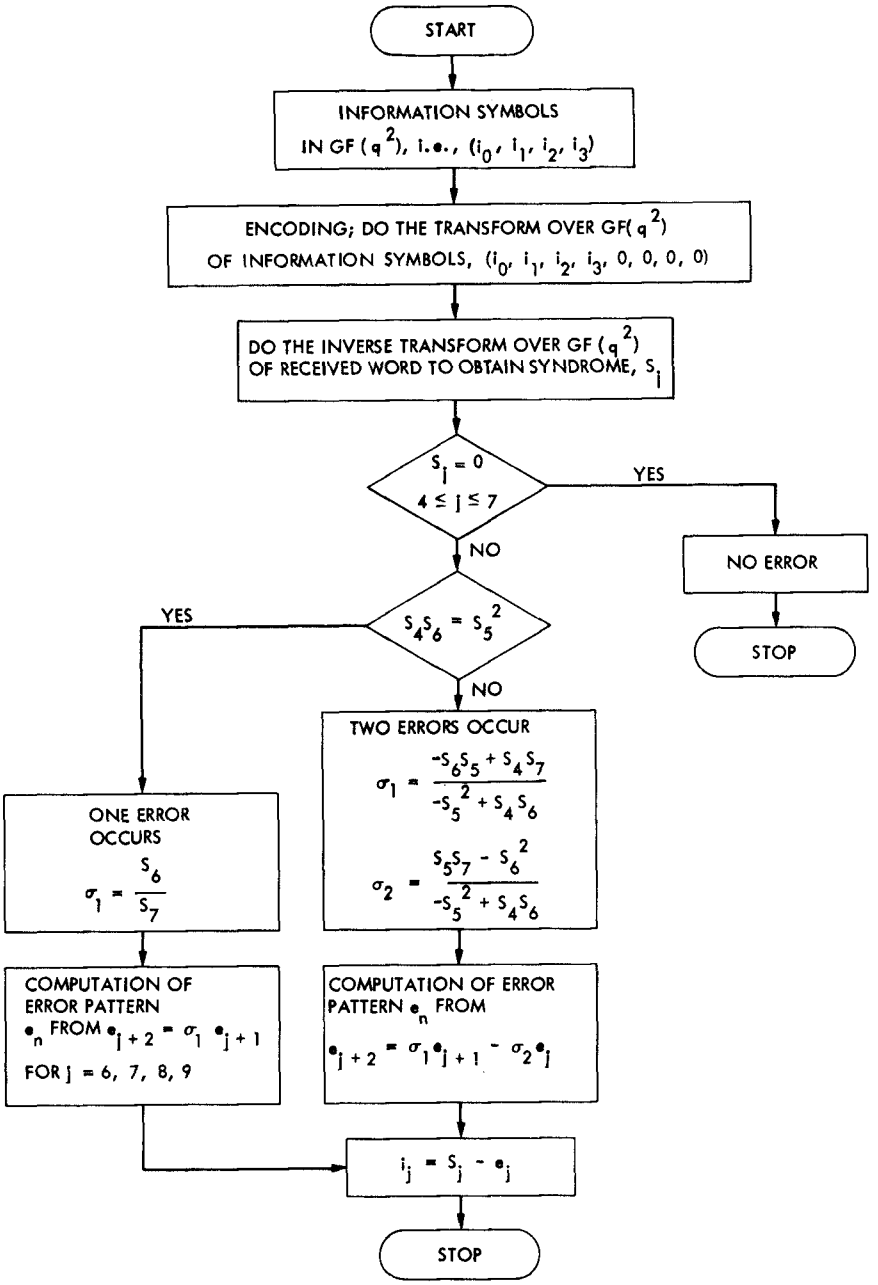

Fig. 1. Transform over GF (q^2) encoder and decoder for correcting at most 2 errors of 8-tuple RS code